Arabic Text Classification Using Deep Learning Technics

Article *in* International Journal of Grid and Distributed Computing · September 2018

DOI: 10.14257/ijgdc.2018.11.9.09

Mohamed Biniz
University Sultan Moulay Slimane
25 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Samir Boukil
Université Chouaib Doukkali
4 PUBLICATIONS **65** CITATIONS
SEE PROFILE

Loubna Cherrat
Encgt
34 PUBLICATIONS **99** CITATIONS
SEE PROFILE

International Journal of Grid and Distributed Computing
Vol. 11, No. 9 (2018), pp.103-114
http://dx.doi.org/10.14257/ijgdc.2018.11.9.09

Arabic Text Classification Using Deep Learning Technics

Samir Boukil[1], Mohamed Biniz[2*], Fatiha El Adnani[3], Loubna Cherrat[4] and
Abd Elmajid El Moutaouakkil[5]

[1,4,5]Laboratory (LAROSERI), Computer Science Department,
Chouaïb Doukkali University, El Jadida, Morocco
[2]University Sultan Moulay Slimane, Faculty of Sciences and Technics
Beni Mellal, Morocco
[3]Laboratory of Mathematics Applied on Physics and Industry, Mathematics
Department, Chouaïb Doukkali University, EL Jadida, Morocco
[1]boukilsamir@yahoo.fr, [2]mohamedbiniz@gmail.com, [3]ftheladnani@gmail.com,
[4]cherratloubna2@gmail.com, [5]elmsn@hotmail.com

Abstract

Text classification is the process of gathering documents into classes and categories based on their contents. This process is becoming more important due to the huge textual information available online. The main problem in text classification is how to improve the classification accuracy. Many algorithms have been proposed and implemented to solve this problem in general. However, few studies have been carried out for categorizing and classifying Arabic text. Technically, the process of text classification follows two steps; the first step consists on selecting some special features from all the features available from the text by applying features selection, features reduction and features weighting techniques. And the second step applies classification algorithms on those chosen features. In this paper, we present an innovative method for Arabic text classification. We use an Arabic stemming algorithm to extract, select and reduce the features that we need. After that, we use the Term Frequency-Inverse Document Frequency technique as feature weighting technique. And finally, for the classification step, we use one of the deep learning algorithms that is very powerful in other field such as the image processing and pattern recognition, but still rarely used in text mining, this algorithm is the Convolutional Neural Networks. With this combination and some hyperparameter tuning in the Convolutional Neural Networks algorithm we can achieve excellent results on multiple benchmarks.

Keywords: *Text mining; Term Frequency-Inverse Document Frequency; Deep learning; Convolutional Neural Network; Classification; Categorization; Natural Language Processing; Arabic language*

1. Introduction

Arabic language is one of the most widely spoken languages in the world. It is the fifth most spoken language in the word and the fifth most used language in the internet. More than 6.6% of the world's population speaks Arabic language (more than 422 million speakers).

The complexity level of applications processing digital Arabic documents and Arabic text is growing rapidly. Thus, the need for organizing these resources so the applications could be more efficient and more productive [1]. Text classification is the process of assigning one or more predefined classes or categories to the analyzed document, based

Received (May 5, 2018), Review Result (August 23, 2018), Accepted (September 4, 2018)

[*] Corresponding Author

on its content. Various classification algorithms were tested on Arabic text classification, But, the majority of them suffer from two big issues: the high dimensionality of the feature space and the rate of the precision is approximately low.

While extracting features from the text document, the features extracted does not all represent semantically the document, and many of them have no influence on the significance of the document. Therefore, the effective determination of feature words is a crucial operation in text classification, this operation contains a series of operations: feature extraction, feature selection and feature weighting. Feature extraction extracts all the possible features from the document, feature selection selects and choose the right and the most significant features from the document and feature weighting is a way to attribute a weight to the words to define how significant that feature is.

In addition to the two problems mentioned in before, we can mention another problem concerning the classification algorithm itself. The complexity of many learning algorithms increases in parallel with the increase in data dimension. Wherefore, algorithms that can improve the classification efficiency, by decreasing the dimensional space of data, are deeply preferred [2]. Those algorithms are widely used in other fields such as image processing and pattern recognition, but rarely used in text mining.

In this paper, we present a solution of the text classification problem in Arabic language. This solution is based on a combination of one of the most interesting techniques of vector-words presentation, which is the Term Frequency-Inverse Document Frequency (TF-IDF), with the Convolutional Neural Network (CNN) one of the most famous deep learning algorithms used especially in image processing and pattern recognition fields. We firstly use a web-crawler to build a 319 million Arabic words corpus, and we apply the TF-IDF technique to produce Arabic word representations using this corpus. Finally, a CNN model trained on top of this corpus is used for Arabic text classification. The simulation results show that the proposed scheme has a better performance than existed approaches for the Arabic text classification on different benchmarks (The recall, f-measure and precision).

This paper is organized as follows: Introduction has been presented in Section 1. Section 2 presents the Arabic language and the different steps of text classification. Section 3 describes a state of art about feature extraction techniques and classification of Arabic text document. Section 4 introduces some important background to facilitate the understanding the following sections. Section 5 explains how our system works. Experiments and results are discussed in section 6 and a conclusion is cited in Section 7.

2. Arabic Language and Classification

2.1. Arabic Language

Arabic Language is one of the most spread languages. It is the 5th most used language in the word and the 5th most used language in the internet. Moreover, the Arabic language is used by more than 422 million people in 2017: 290 million as first language and 132 million as second language.

The Arabic language has a special way for writing. In the contrary of English, French and all the western languages in general, the Arabic language is writing from the right to the left, and there is no lowercase and uppercase in Arabic letters. The shapes of some letters can change according to their position in the word, for example the letter “ع” can be writing in four different ways depending on his place in the word: in the beginning of the word “box, علبة : عـ”, in the middle of the word “game, لعبة: ـعـ” and in the end of the word “Square, مربع : ـع | radio, مذياع : ع” . It is one of the most challenging languages in the world with its rich morphology, its complex syntax, and its difficult semantics. This makes its analysis and automatic processing very hard and complex.

2.2. Text Classification Steps

Generally, text categorization process includes five main steps: [3]. And a preliminary step concerning gathering and preparing the corpus on which the work will be done could be added as a 6th step.

1) Document Preprocessing: In this step, numbers, symbols, rare words and stop words are removed, and some stemming is applied, this can be done easily in French, English or Spanish, but it is more difficult in Arabic. For Arabic language there are several stemming methods [4] that we can apply in the preprocessing step, but the most of this methods are based on two approaches: morphological analysis of the words (root-stemming and light-stemming techniques) [4] and statistical analysis (N-gram technique) [5].

2) Document representation: Before classification, documents have to be represented in a format that the classification algorithm can recognize, Bag Of Words (BOW) is one of the most used methods. It is a representation of text that describes the occurrence of words within a document.

3) Dimension Reduction: There are, sometimes, hundreds of thousands of words in a document, so it is not possible to do the classification for all those words as features; also, the computer could have problems processing such amount of data. That is why it is important to select the most representative features as inputs for the classification step.

4) Model Training: This is the main part of text categorization. It includes choosing a part of documents from the corpus to establish the training set, performs the learning on it, and after that generates the model.

5) Testing and Evaluation: This step uses the model generated from the previous step, and performs the classification on a part of the corpus called the testing set. Next, we choose appropriate index to do the evaluations of the model.

3. Related Works

Numerous researches have been focused on text classification in English, French and even Spanish. Even though, works on text classification for Arabic language remain restricted [6]. Among those works, several recent researches have been proposed, we mention:

Mesleh [7] produced a text classification system for Arabic language documents. The achieved system uses 1) CHI statistics as a feature extraction method in the pre-processing step, and 2) Support Vector Machines (SVM) classification algorithm for text classification tasks. The corpus was gathered from online Arabic newspaper archives in addition to some other websites. This corpus contains 1445 documents classified into 9 categories. Experimental results indicate a high classification efficiency in term of F-measure compared to other classification algorithms.

Al-Harbi *et al.*, [6] introduced Arabic document classification using statistical methodology. The classification was performed on seven diverse Arabic corpora. The performance of two well-known classification algorithms (SVM and C5.0) in classifying the seven Arabic corpora has been evaluated. Globally, C5.0 classifier demonstrates the best classification accuracy.

Noaman *et al.*, [8] presented the utilization of Naïve Bayes classifier with rooting algorithm to classify Arabic document. To approve the proposed algorithm, the authors created a corpus of 300 documents categorized into 10 classes. The corpus was collected from many newspaper articles gathered from various online newspaper websites. The experimental study demonstrates the achievement of the proposed classifier in terms of error rate, recall measures, and accuracy, it accomplishes 62.23% of classification accuracy.

Alsaleem [9] examined the problem of automatic categorization of Arabic text documents and utilized two different algorithms, Support Vector Machine algorithm (SVM) and Naïve Bayesian method (NB), on various Arabic datasets to deal with the Arabic text categorization problem. The Experimental outcomes using different Arabic text categorization datasets affirm that SVM algorithm surpass the NB with regards to F1, Recall and Precision measures.

Kourdi *et al.*, [10] utilized statistical machine learning algorithm Naive Bayes (NB) to categorize Arabic web documents. They collected own datasets classified into five classes comprise of 300 documents for each class. Their technique results demonstrated that the average accuracy was 62%.

Hmeidi *et al.*, [11] depict study of Arabic text classification utilizing two machine learning algorithms: The K nearest neighbor (KNN) and support vector machines (SVM). They make their own corpus by collecting news articles for training and testing steps. They demonstrated that these algorithms results are most effective however the SVM algorithm was better in prediction.

Mountassir *et al.*, [12] led a binary sentiment classification from Arabic text documents employing three classifiers: NB, SVM and KNN. Two corpora were utilized: the first is created by these authors and is made from two particular datasets (movies and sports). The second is OCA, a corpus of movie surveys created by Rushdi-Saleh *et al.*, [13]. Before the classification step, the authors executed a preprocessing step by extirpating stop words, isolating words from their clitics, eliminating terms used less than three times in the dataset, and by replacing words by their stems. The authors discovered that pre-processing, n-grams combination, and presence-based weighting enhance the classification performance.

N. Boudad *et al.*, [14] conducted a comparative study about sentiment analysis based on Arabic text documents. They studied more than twenty works, and they showed that work based on stemming as feature selection techniques and SVM for classification gave better accuaracy.

4. Background

4.1. Term Frequency-Inverse Document Frequency

Here, we will introduce a description of the TF-IDF algorithm. Basically, TF-IDF works by determining the relative frequency of words in a document compared to the inverse ratio of that word over the whole corpus. Naturally, this calculation determines how important a given word is in a specific document. Words that are common in a single or few documents are apt to have higher TF-IDF numbers than common words, for instance, articles and prepositions. Given a group of documents *D*, a word *w*, and an individual document $d \epsilon D$, we calculate w_d the weight of the word *w* by applying (1) and (2).

$$w_d = TF * IDF \tag{1}$$

$$w_d = f_{w,d} * log\ (|D| / f_{w,D}) \tag{2}$$

where $f_{w,d}$ is the number of times the word *w* appears in the document *d*, $|D|$ is the size of the corpus, and $f_{w,D}$ is the number of documents in which *w* appears in *D* [15].

Suppose that $|D| \sim f_{w,D}$, *i.e.*, the size of the corpus is roughly equivalent to the frequency of *w* over *D*.

$$1 < log\ (|D|/ fw, D) < cte \tag{3}$$

If (3) is true for some very small constant *cte*, at that point w_d will be smaller than $f_{w,d}$ yet still positive. This suggests that *w* is approximately common over the whole corpus yet at the same time holds some significance throughout *D*. For instance, this could be the situation for extremely basic words such as articles, pronouns, and prepositions, which hold no relevant meaning by themselves. Such common words consequently get a low TF-IDF score, rendering them basically irrelevant in the search.

Finally, imagine that $f_{w,d}$ is large and $f_{w,D}$ is small. At that point $log\ (|D|/\ f_{w,D})$ will be rather large, and so w_d will evenly be large. This is the situation we are most interested in, since words with high w_d suggest that the word *w* is an important word in the document *d* but not common in the corpus *D*. This *w* word is said to have an important discriminatory power.

4.2. Convolutional Neural Networks

Most of classification algorithms require the construction of a multidimensional feature vector used as input to the algorithm. Therefore, experts become indispensable to determine the feature vectors of the desired operation. A different and innovative approach lies in not using an expert for the construction of the feature vector, by automatically extracting it using a learning algorithm. Deep learning algorithms have the ability to automatically learn features useful for the classification task. Convolutional neural networks (CNN) are variants of those deep learning algorithms.

Convolutional Neural Networks are very similar to ordinary Neural Networks, they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, execute a dot product and it may follow it with a non-linear function. The entire system still expresses a single differentiable score function: from the inputs to the classes in the outputs. And they still have a loss function on the last layer.

CNN is typically a sequence of layers, and the outputs of every layer is the inputs of the next layer. Those layers are: convolutional layers, pooling layers and fully connected layers. [16, 17].

5. Arabic Text Classification With Deep Learning Algorithms

In this work, Arabic text classification was performed using the CNN algorithm. Considering that the aim of our work is to benefit from the advantages of that algorithm, which were proved in other fields, in the Arabic text.

Figure 1, presents the principle modules in our system; the next sections describe each of these modules.

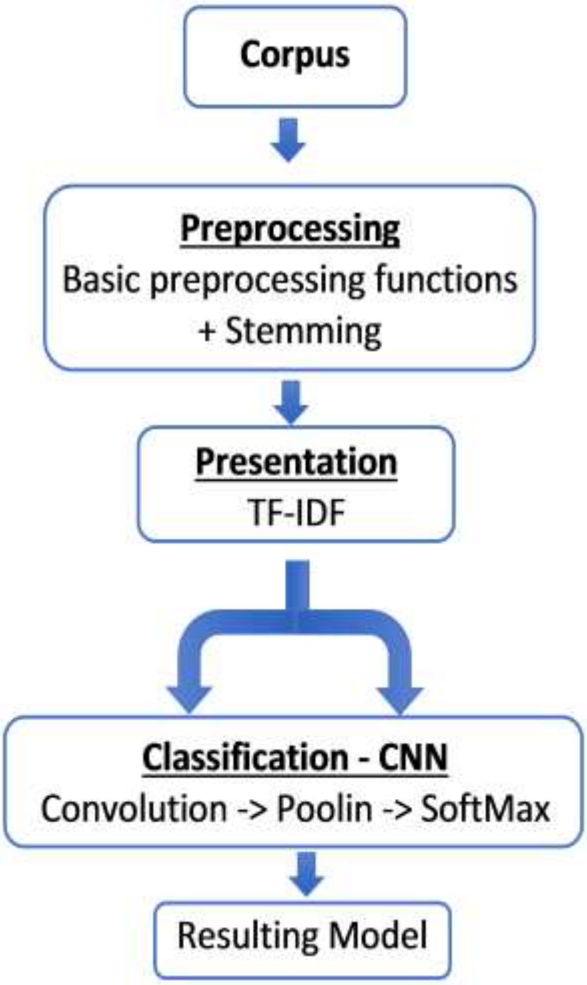


Figure 1. System Architecture

5.1. Document Pre-Processing

First, we applied basic functions for preprocessing, which includes: removing stop words, removing foreign characters, removing punctuation, removing articles and numbers. Then, we execute the stemming algorithm [4], which transforms all words to their stems.

5.2. Document Representation and Dimension Reduction

In this step, we applied the TF-IDF algorithm. Every stem was replaced with a score that expresses the importance of that stem in the entire corpus. Stems with very low scores were eliminated.

5.3. Classification: Convolutional Neural Networks

In our proposed CNN model, the typical convolutional layer, pooling layer and fully-connected layer are included, which is shown in Figure 2.

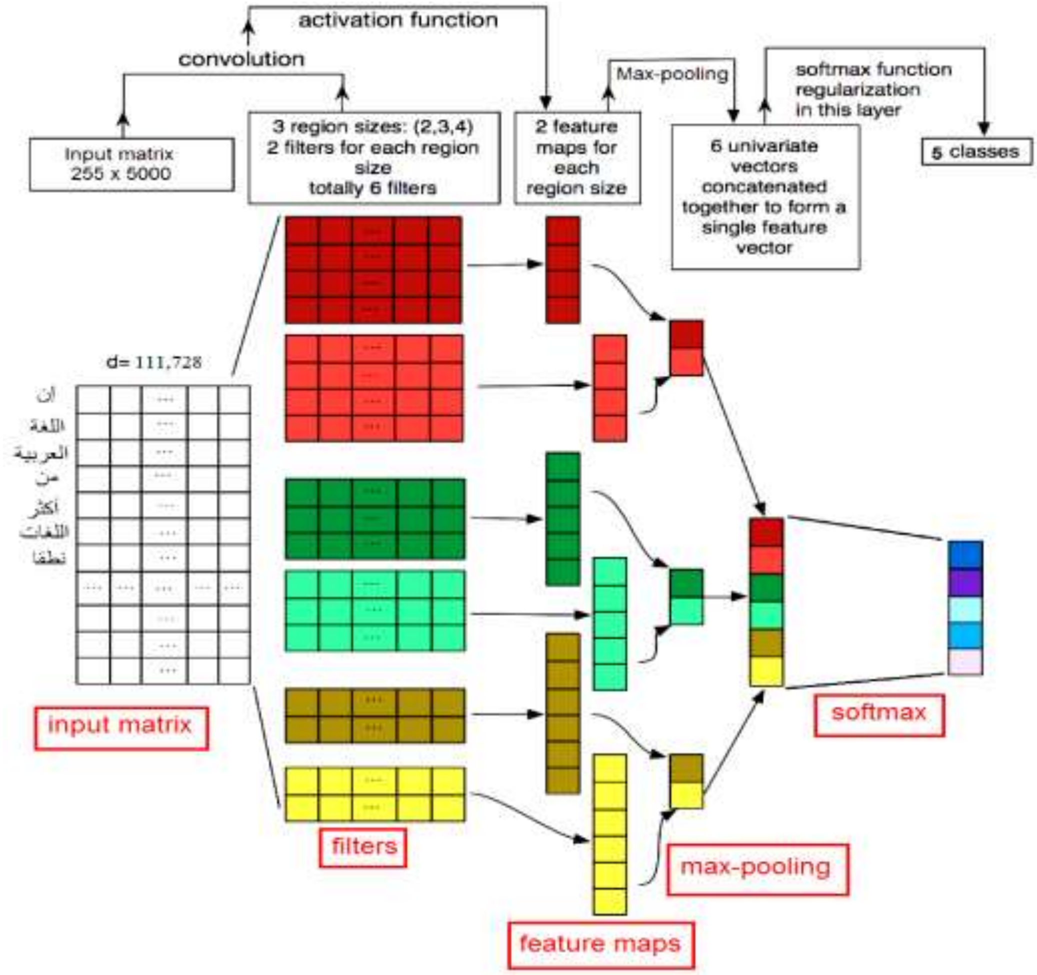


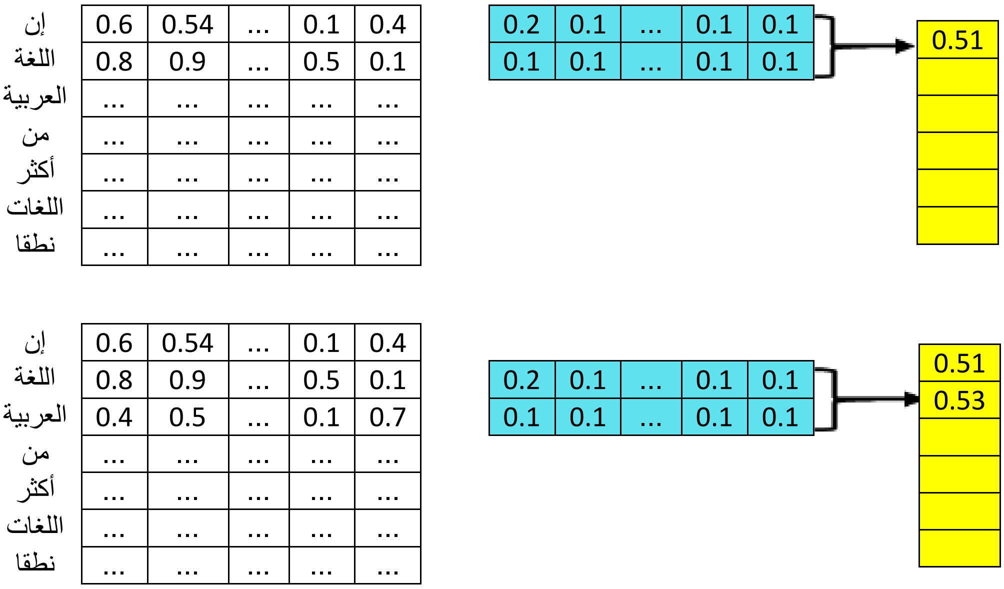
Figure 2. Illustration of a Convolutional Neural Network (CNN) Architecture for Arabic Text Classification

We used a one-layer CNN on 255 words, with word vector of dimension 111,728.

1) Input matrix: We fixed the input in 255 words, and that's because 255 is the maximum number of stems included in any document in the corpus after the pre-processing step. The dimension of the word vectors is 111,728, and that is the total number of documents in the whole corpus. We let *s* mean the number of words and *d* mean the dimension of the word vector, therefrom we now have an input matrix of the shape *s* x *d* or 255 x 111,728.

2) Filters: One of the desirable properties of CNN is that it preserves orientation, that's good for us because texts have a one-dimensional structure where words sequence matter. We also recall that all words in the example are each replaced by a 111,728-dimensional word vector. Therefrom, one dimension of the filter was fixed for all the filters to match the word vectors, and the region size "h" was varied from one filter to another. Region size refers to the number of rowsof the input matrix representing words. In the Figure 2. filters are the illustrations of the filters, not the results of filtering the input matrix. We picked here to utilize 3 region sizes, each region recovers respectively 2, 3 and 4 words (stems) at once. Furthermore, 2 filters for each region was chosen. Generally speaking, there is 6 filters.

3) Feature maps: In this section, we explain how convolutions / filtering are performed by the CNN. We have filled the input matrix with some results obtained from the pretreatment step, and the filter matrix also is filtered with some numbers for clarity the explanation. This example is illustrated in Figure 3.



إن	0.6	0.54	...	0.1	0.4
اللغة	0.8	0.9	...	0.5	0.1
العربية	...	...	...	...	...
من	...	...	...	...	...
أكثر	...	...	...	...	...
اللغات	...	...	...	...	...
نطقا	...	...	...	...	...

0.2	0.1	...	0.1	0.1
0.1	0.1	...	0.1	0.1

0.51

إن	0.6	0.54	...	0.1	0.4
اللغة	0.8	0.9	...	0.5	0.1
العربية	0.4	0.5	...	0.1	0.7
من	...	...	...	...	...
أكثر	...	...	...	...	...
اللغات	...	...	...	...	...
نطقا	...	...	...	...	...

0.2	0.1	...	0.1	0.1
0.1	0.1	...	0.1	0.1

0.51
0.53

Figure 3. How Convolutions (filtering) Work

In the beginning, the two-word filter, represented by the (2x111,728) yellow matrix *w*, overlays over the word vectors of “إن” and “اللغة”. After that, it performs an element-wise product for all its (2x111,728) elements, and then sum them up and get one number (0.6x0.2 + 0.5x0.1 + … + 0.4x0.1 = 0.51). 0.51 is recorded as the first element of the output sequence, *o*, for this filter.

At that point, the filter moves down 1 word and overlays across the word vectors of ‘اللغة’ and ‘العربية’ and perform the same operation to get 0.53. In this manner, *o* will have the dimensions (*s*–*h*+1 x 1), in this case (254x1).

To get the feature map, *c*, we add a *bias* term and apply an activation function, we chose the Rectified Linear Unit function (ReLU). This gives us *c*, with the same dimensions as *o* (254 x 1).

4) max-pooling: The dimensionality of c is reliant both s and h, in other words, it will change through filters of various region sizes. To handle this issue, we used the max-pooling function and extract the biggest number from each c vector.

5) softmax: After max-pooling, the resulting vector have a fixed-length of 6 elements (number of filters). This resulting vector can then be fed into a fully-connected (softmax) layer to perform the classification task. As part of the learning process, the error from the classification is back-propagated into the following parameters:

- The *w* matrices that resulted *o*
- The bias term that is added to *o* to produce *c*

5.4. Result

The resulting model is ready to use for classifying new instances.

6. Experiment and Results

The work has been done on a MacBook Pro i7 6th Generation, RAM 16 Go, Hard disk: SSD 512 Go, touring on MacOs High Sierra.

6.1. Datasets

The dataset is a collection of Arabic texts, which covers modern Arabic language used in newspapers articles. The text contains alphabetic, numeric and symbolic words. The dataset consists of 319,254,124 words (cf. Table. I) and 111,728 documents (cf. Table. II) structured in text files and collected from 3 Arabic online newspapers: Assabah (www.assabah.ma),Hespress (www.hespress.com) and Akhbarona (www.akhbarona.com) using semi-automatic web crawling process.

The documents in the dataset are categorized into 5 classes: sport, politic, culture, economy and diverse. The number of documents and words for each class varies from one class to another (cf. Tables I – II).

This dataset was divided into two sections: training dataset and testing dataset. The training dataset represents 70% of each class and it allows us to build the model, while the testing dataset represents the remaining 30% and it helps us to verify the accuracy of our model. Records in the dataset were stored as matrices, where every word is presented as a vector.

Table 1. Number of Words in the dataSet

Site	sport	politic	culture	economy	diverse	Total
Assabah	90,182,142	10,693,166	19,795,393	9,309,981	20,402,016	150,382,699
Hespress	17,728,154	22,771,705	11,423,641	13,567,983	12,078,694	77,570,177
Akhbarona	9,465,337	33,293,023	10,997,014	19,543,260	18,002,614	91,301,248
TOTAL	**117,375,633**	**66,757,894**	**42,216,048**	**42,421,224**	**50,483,325**	**319,254,124**

Table 2. Number of Documents in the dataSet

Site	sport	politic	culture	economy	diverse	Total
Assabah	34,244	2,381	5,635	2,620	9,253	**54,133**
Hespress	6,965	5,737	3,023	3,795	7,475	**26,995**
Akhbarona	5,313	12,387	5,080	7,820	NAN	**30,6**
TOTAL	**46,522**	**20,505**	**13,738**	**14,235**	**16,728**	**111,728**

6.2. CNNs Experiments Setup

The choice of hyper-parameters of CNNs model may influence the performance of the classification. In our experiments we employ stochastic gradient descent (SGD) to train the network and use backpropagation algorithm to calculate the gradients. We initialize the learning rate at 0.001. Likewise, we utilize Dropout [3] to enable the model to better converge (dropout ratio as 0.5). We directed the next experiments to select two important parameters: filter sizes and feature maps.

1) Influences of Filter Sizes: We initially execute a search across a single filter size to locate the "best" size for the dataset. From Table. III. we can say that the best filter sizes combination is (1,2,3,5,7,9). Yet, the training time for this filter combination is considerably big: 60 hours. Considering the efficiency and performance we choose (2,3,4) in our model.

2) Influences of Feature Maps: Table. IV. demonstrates that expanding the number of feature maps over 400 usually harms the efficiency (probably due to overfitting). Another remarkable reasonable point is that it takes more time to train the model when the number of feature maps is expanded. We choose 100 in our last model.

Table 3. Influences of Filter Sizes (number of feature maps fixed)

Filter sizes	Test accuracy	Training time (hours)
(1)	84.4	2.5
(2)	85.7	4
(3)	85.9	6
(4)	86.1	10
(1,2)	86.7	6
(2,3)	86.8	10
(1,2,3)	86.6	18
(2,3,4)	**87.2**	12
(2,3,5,7)	87.3	35
(1,2,3,5,7,9)	**88.8**	60

Table 4. Influences of Feature Maps (filter size fixed at (2.3.4))

Feature maps	Test accuracy	Training time (hour)
50	85.8	05
100	87.2	1
200	87.8	2.5
400	88.1	5
800	87.5	10.5
1000	87.3	11

6.3. Baseline Methods

The methods that we take, in this works, as baseline methods for Arabic text classification are: Logistic Regression (LR) and Support Vector Machine (SVM)

6.4. Results and Discussions

To test the influence of the size of the datasets, we trained models on different sizes of the datasets (data_27k, data_55k, data_83k, data_111k), the number of documents in each size of the dataset is illustrated in Table. V. From Table VI we can see that the larger the size of the dataset we use, the better accuracy we get. We can notice that CNNs model realize the best results for all the sizes. It demonstrates that CNNs model can effectively compose the semantic representation of texts and catch more contextual information of features compared with traditional methods based on bag-of-words model.

Besides, the training and testing data are drawn from various distribution and sources, and in many cases the documents can be classified in multiple classes. Considering these challengeable cross-domain classification tasks, the accuracy over 92 % is more than satisfying.

Table 5. Different Sizes of the Dataset

Dataset	Size (number of document)
data_27k	27,932
data_55k	55,864
data_83k	83,796
data_111k	111,728

Table 6. Accuracy of CNNs, LR and SVM on Different Training Datasets

	data_27k	data_55k	data_83k	data_111k
LR	81.23	82.14	83.46	86.31
SVM	83.04	84.77	86.90	88.20
CNNs	86.30	87.12	89.75	92.94

7. Conclusion

In this paper we propose a simple and efficient method to classify Arabic text from large dataset. We assess our dataset with CNNs model and some traditional machine learning models as benchmark. We reason that CNNs model has better performance on the task of Arabic text classification. At the point when the dataset is large and big, traditional technique, for example, SVM, cannot accomplish great performance as good as CNN model.

References

[1] R. F. Correa and T. B. Ludermir, “Automatic Text Categorization: Case Study”, Proceedings of the 7th Brazilian Symposium on Neural Networks, Pernambuco, Brazil, **(2002)** November 11-14 .

[2] J. Yan, N. Liu, B. Zhang, S. Yan, Z. Chen, Q. Cheng, W. Fan and W. Y. Ma, “OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization”, Proceedings of the 28th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, **(2005)** August 15-19.

[3] F. Al-Zaghoul and S. Al-Dhaheri, “Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks”, Proceedings of the 15th International Conference on Computer Modelling and Simulation (UKSim), Cambridge University, United Kingdom, **(2013)** April, pp. 485-490.

[4] S. Boukil, F. E. Adnani, A. E. E. Moutaouakkil, L. Cherrat and M. Ezziyyani, “Arabic Stemming Techniques as Feature Extraction Applied in Arabic Text Classification”, Proceedings of the International Conference on Advanced Information Technology, Services and Systems (AIT2S-17), Tangier, Morocco, **(2017)** April 14-15, pp. 349–361.

[5] N. Yousef, A. Abu-Errub, A. Odeh and H. Khafajeh, “An Improved Arabic Word’s Roots Extraction Method Using N-Gram Technique”, Journal of Computer Science., vol. 10, no. 4, **(2014)**, pp. 716-719.

[6] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M.S. Khorsheed and A. Al-Rajeh, “Automatic Arabic Text Classification”, Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data, Lyon, France, **(2008)** March 12-14.

[7] A. M. A. Mesleh, “Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System”, Journal of Computer Science., vol. 3, no. 6, **(2007)**, pp. 430-435.

[8] H. M. Noaman, S. Elmougy, A. Ghoneim and T. Hamza, “Naive Bayes Classifier based Arabic document categorization”, Proceeding of the 7th International Conference on Informatics and Systems (INFOS2010), Cairo, Egypt, **(2010)** March 28-30, pp. 1-5.

[9] S. Alsaleem, “Automated Arabic Text Categorization Using SVM and NB”, International Arab Journal of e-Technol., vol. 2, No. 2, **(2011)** June, pp. 124-128.

[10] M. El-Kourdi, A. Bensaid and T. Rachidi, “Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm”, Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, Switzerland, **(2004)** August 28-28, pp. 51-58.

[11] I. Hmeidi, B. Hawashin and E. El-Qawasmeh, “Performance of KNN and SVM Classifiers on Full Word Arabic Articles”, Journal of Advanced Engineering Informatics., vol. 22, no. 1, **(2008)**, pp. 106-111.

[12] A. Mountassir, H. Benbrahim and I. Berrada, “An empirical study to address the problem of unbalanced data sets in sentiment classification”, Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, South Korea, **(2012)** October 14-17, pp. 3298–3303.

[13] M. Rushdi-Saleh, M.T. Martín-Valdivia, L.A. Ureña-López and J.M. Perea-Ortega, “OCA: opinion corpus for Arabic”, Journal of the American Society for Information Science and Technology., vol. 62, no. 10, **(2011)** October, pp. 2045–54.

[14] N. Boudad, R. Faizi, R. Oulad Haj Thami and R. Chiheb, “Sentiment analysis in Arabic: A review of the literature”, Ain Shams Engineering Journal., Available online **(2017)** July 21.

[15] G. Salton and C. Buckley, “Term-weighing approache in automatic text retrieval”, Information Processing & Management Journal., vol. 24, no. 5, **(1988)**, pp 513-523.

[16] A. Krizhevsky , I. Sutskever and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), Lake Tahoe, Nevada, **(2012)** December, vol 1, pp. 1097-1105.

[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, vol. 86, no. 11, **(1998)** November, pp. 2278-2324.